Journal of Computational Physics **152,** 385–403 (1999)
Article ID jcph.1999.6255, available online at http://www.idealibrary.com on IDEAL®

An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients

C. Vuik,* A. Segal,* and J. A. Meijerink†

*Department of Technical Mathematics and Informatics, Faculty of Information Technology and Systems, Delft University of Technology, Mekelweg 4, 2628 CD Delft, the Netherlands; and †Shell International Exploration and Production, P.O. Box 60, 2280 AB Rijswijk, the Netherlands
E-mail: c.vuik@math.tudelft.nl

Received May 1, 1998; revised November 17, 1998

Knowledge of fluid pressure is important to predict the presence of oil and gas in reservoirs. A mathematical model for the prediction of fluid pressures is given by a time-dependent diffusion equation. Application of the finite element method leads to a system of linear equations. A complication is that the underground consists of layers with very large differences in permeability. This implies that the symmetric and positive definite coefficient matrix has a very large condition number. Bad convergence behavior of the CG method has been observed; moreover, a classical termination criterion is not valid in this problem. After diagonal scaling of the matrix the number of extreme eigenvalues is reduced and it is proved to be equal to the number of layers with a high permeability. For the IC preconditioner the same behavior is observed. To annihilate the effect of the extreme eigenvalues a deflated CG method is used. The convergence rate improves considerably and the termination criterion becomes again reliable. Finally a cheap approximation of the eigenvectors is proposed.

Key Words: porous media; preconditioned conjugate gradients; deflation; Poisson equation; discontinuous coefficients across layers; eigenvectors; finite element method.

1. INTRODUCTION

One of the problems an oil company is confronted with when drilling for oil is the presence of high fluid pressures within the rock layers of the subsurface. Knowledge of the fluid pressures is important to predict the presence of oil and gas in reservoirs and is a key factor in the safety and environmental aspects of drilling a well.

0021-9991/99 $30.00

A mathematical model for the prediction of fluid pressures on a geological time scale is based on conservation of mass and Darcy's law ([6, 14]). This leads to a time-dependent diffusion equation, where the region also changes in time as rocks are deposited or eroded. The Euler backward method is used for the time integration. In order to solve this diffusion equation, the finite element method is applied. As a consequence, in each time-step, a linear system of equations has to be solved. Due to nonlinear effects and the time-dependence of the region the coefficients of the diffusion equation change in each time-step.

In practical applications we are faced with large regions in a three-dimensional space and as a consequence a large number of finite elements are necessary. The matrix itself is sparse, but due to fill-in a direct method requires too much memory to fit in core. Moreover, since in each time-step we have a good start vector, only iterative methods are acceptable candidates for the solution of the linear systems of equations.

Since these equations are symmetric a preconditioned conjugate gradient method (ICCG) [26] is a natural candidate. Unfortunately, an extra complication of the physical problem we are dealing with is that the underground consists of layers with very large differences in permeability. For example, in shale the permeability is of order 10^{-6} to 10^{-11} (D), whereas in sandstone it is of order 1 to 10^{-4} (D). Hence a contrast of 10^{-7} is common in the system of equations to be solved. Other applications where the coefficients have large discontinuities are electrical power networks [21], groundwater flow [1, 18], semiconductors [11], and electromagnetics modeling [19].

A large contrast in coefficients usually leads to a very ill-conditioned system of equations to be solved. Since the convergence rate of ICCG depends on the distribution of the eigenvalues of the matrix one may expect a slow convergence rate. In Section 2 it is shown that this is indeed the case. An even more alarming phenomenon is that numerical results suggest that ICCG has reached a certain accuracy but that the actual accuracy is in fact orders of magnitude worse. This is due to the ill-conditioned matrix, which results in the standard termination criterion no longer being reliable. To our knowledge, this observation has not been made before.

An analysis of the problem in Section 3 shows that without preconditioning there are many small eigenvalues in the matrix, but using a diagonal preconditioned matrix this number is reduced to the number of sandstone layers that do not reach the earth's surface. This analysis suggests a way of solving the problems mentioned. In Section 4 it is shown that the convergence and reliability of the termination criterion is considerably improved by projecting the solution in each iteration onto the orthogonal complement of the space spanned by the eigenvectors corresponding to the very small eigenvalues of the preconditioned matrix. The idea is that, assuming that the vectors are written as linear combination of eigenvectors, the components corresponding to these specific eigenvectors do not play a role any more. As a result, one may expect much faster convergence and a reliable termination criterion. A clear disadvantage of this method is of course that one has to compute the specific eigenvectors. In Section 5, however, it is shown how one can approximate these eigenvectors easily, based on physical arguments. Furthermore, it is shown that even approximate eigenvectors lead to fast convergence. Finally, in Section 6, some numerical evidence on our improved algorithm is given.

The CG [20] method, combined with a preconditioner, is a popular iterative method for solving large algebraic systems of linear equations, when the system matrix is symmetric and positive definite. Many practical preconditioners are based on an Incomplete Choleski factorization. The resulting ICCG method was first described in [26]. Various alternatives

have since been formulated, such as MICCG [17], RICCG [4], and ILUM [32]. Recently a number of preconditioners have been proposed for the discretized Poisson equation, where the rate of convergence does not depend on the grid size. Examples are NGILU [37] and DRIC [31]. A comparison of these and related preconditioners has been given in [7].

It is well known that the convergence rate of CG depends on the ratio of the largest and smallest eigenvalues of the matrix. To explain the superlinear convergence of CG, Ritz values have to be taken into account [38]. The convergence rate depends only on active eigenvalues. An eigenvalue is active when the error has a non zero component in the corresponding eigenvector. This observation is used to solve singular systems with the CG method (see [3, pp. 476–480]). In [23] the initial approximation is projected so that the start residual is perpendicular to the kernel of the matrix. In [2] the start approximation is projected in such a way that the error has no components in the eigenvectors corresponding to small eigenvalues. This increases the smallest active eigenvalue and thus the convergence rate. After the projection the original CG method has been used in both papers. An incomplete factorization preconditioner for singular systems has been investigated in [30].

In [28, 29] a deflated CG method was proposed. In every CG iteration the residual is projected onto a chosen subspace. The projected CG method used in this work [40] is closely related to these deflated CG methods. The main difference is the choice of the subspace. We base our choice on the physical properties of the problem considered. Another difference is the implementation. Various implementations are possible to incorporate a projection. We specify an implementation such that the basis of our subspace consist of vectors with many zero elements. Related work has recently been presented in [21, 35].

For the solution of singular non symmetric systems we refer to [8]. Deflation is also used in iterative methods for non-symmetric systems of equations [5, 9, 10, 13, 24, 27, 33]. In these papers the smallest eigenvalues have been shifted away from the origin. The eigenvectors are in general obtained from the Arnoldi method. The motivation to use deflation is to enhance the convergence of restarted GMRES [34]. Finally deflation techniques have also been combined with solution methods for systems of nonlinear equations [36, 39].

2. STATEMENT OF THE PROBLEM AND EXPERIMENTS WITH ICCG

As mentioned in the Introduction, in each time-step we have to solve a system of equations that arises from the discretization of a 3D time-dependent diffusion equation. In this paper, however, we are only interested in the convergence behavior of the ICCG process for problems with layers with large contrasts in the coefficients. For that reason we simplify the equation considerably and assume that we have to solve the stationary linearized 2D diffusion equation in a layered region,

$$-\mathrm{div}(\sigma \nabla p) = 0, \tag{1}$$

with p the fluid pressure and σ the permeability. At the earth's surface the fluid pressure is prescribed. When the pressure field is required in a domain it is not practical to calculate the pressure in every position of the earth's crust. Therefore the domain of interest is restricted artificially. We assume that the lowest layer is bounded by an impermeable layer, so there is no flux through this boundary. The artificial vertical boundaries are taken at a sealing fault, or far away from the reservoir. Again a zero flux condition is a reasonable assumption at these boundaries. For the physical background of this problem we refer to Chap. 12 of [14].

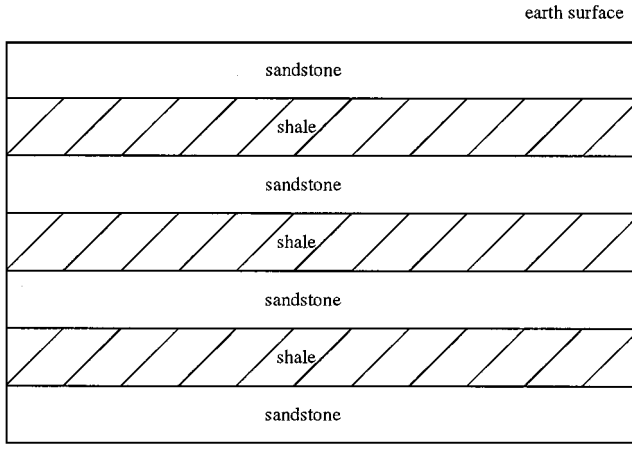

FIG. 1. Artificial configuration with seven straight layers.

For our model problem we assume that σ in sandstone is equal to 1 and σ in shale is equal to 10^{-7}. Furthermore, the Dirichlet boundary condition at the earth's surface is set equal to 1. The solution of Eq. (1) with these boundary conditions is of course $p = 1$, but if we start with $p = 0$ or a random vector, our linear solver will not notice the difference with a real problem. Numerical experiments show that the choice of one of these start vectors has only marginal effects. An advantage of this problem is that the exact error can easily be calculated.

Equation (1) is discretized by a standard finite element method using bilinear quadrilateral elements. This results in a system of linear equations to be solved, which will be denoted as $Ax = b$. In our first experiment we have solved this problem on a rectangular domain with seven straight layers (Fig. 1), using CG without preconditioner. The termination criterion is based on the estimate of the smallest eigenvalue during the iterations by a Lanczos method as described by Kaasschieter [22]. Figure 2 shows the norm of the residual, the

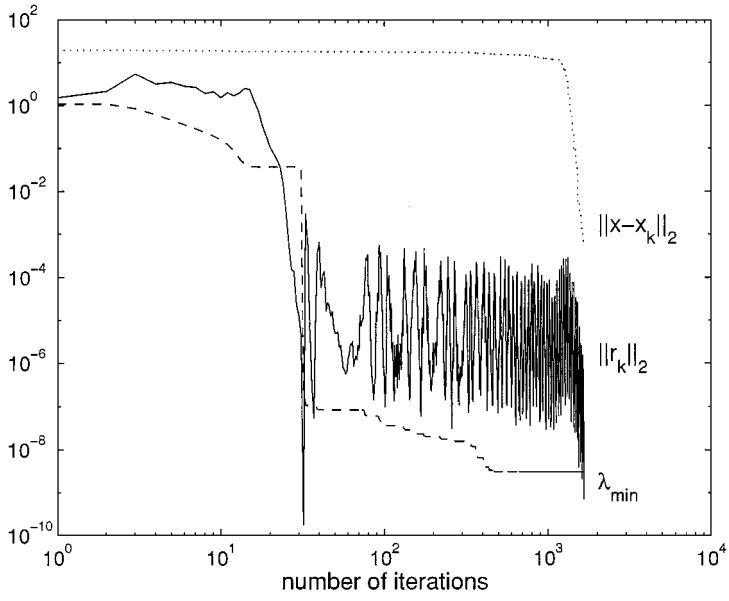

FIG. 2. Convergence behavior of CG without preconditioning.

norm of the true error, and also the estimate of the smallest eigenvalue as function of the number of iterations. In each layer 10 elements in the horizontal and 5 elements in the vertical direction are used. From this figure the following remarkable observations may be made.

1. The residual decreases monotonically between iterations 1 and 30. For the iterations between 31 and 1650 we have an erratic behavior of the residual. After iteration 1650 we again have a monotone decreasing of the residual.

2. If we required an accuracy of order 10^{-2}, the process would stop after approximately 25 iterations, since then the residual divided by the estimate of the smallest eigenvalue is small enough. Unfortunately the true error ($\|x - x_k\|_2$) is still large. The estimated error is not sharp, because the estimate of the smallest eigenvalue is very inaccurate. Since the fluid pressure is used in the prediction of the presence of oil and gas in reservoirs the true error should be small.

3. In iterations 1–30 it looks as if the smallest eigenvalue is of order 10^{-2}, whereas from iteration 31 it is clear that the smallest eigenvalue is of order 10^{-7}.

So we see that the bad condition leads to a large number of iterations. Moreover, for practical values of the error, the termination criterion is not reliable.

Repeating the same experiment using an IC preconditioning gives a drastic reduction of the number of iterations, but still the same conclusions as for the case without preconditioning can be drawn. Figure 3 shows the convergence behavior. Note that the horizontal scales in Figs. 2 and 3 are quite different. Although the number of iterations (48) is small compared to the number for the nonpreconditioned algorithm (1650), still it is quite large compared to the number of unknowns (385). Note that the norm of the true error does not decrease below $cK_2(A)u\|x - x_0\|_2$, where c is a small constant and $u(\approx 10^{-16})$ is the unit round off.

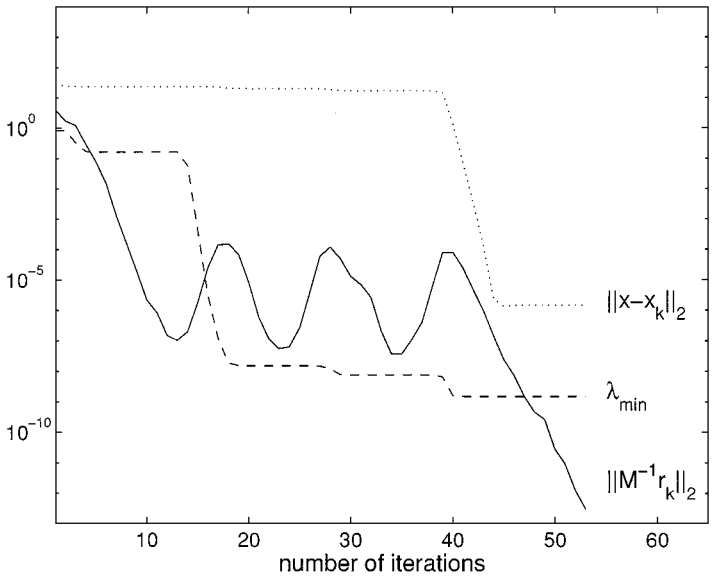

FIG. 3. Convergence behavior of CG with IC preconditioning ($M = L^{\mathrm{T}}L$).

3. ANALYSIS OF THE ITERATION MATRIX

In order to gain more insight into the convergence behavior, we have investigated the eigenvalues of the matrix. If we compute all eigenvalues of the discretization matrix, then we see that the number of small eigenvalues (i.e., of order 10^{-7}), is equal to the number of nodes that are entirely in the shale layers, plus 3. One can expect that this number is at least equal to the number of internal "shale" nodes, since all nonzero elements in the corresponding rows of the matrix are of order 10^{-7}. The number 3 will be explained later on. The iteration process only converges, once all small eigenvalues have been "discovered."

When we use an IC preconditioner, and compute all eigenvalues of the discretization matrix multiplied by the preconditioning matrix, we see that only three eigenvalues are of order 10^{-7}. All other eigenvalues are of order 1. This observation appears to be true for all kinds of preconditioners, even for a simple diagonal scaling. The convergence behavior shown in Fig. 3 can be explained by these three eigenvalues. Once a small eigenvalue has been "discovered" by the CG process, the residual increases considerably. Only when all small eigenvalues are approximated by the Krylov subspace does the true error decrease.

A possible explanation for the fact that there are only three small eigenvalues in the preconditioned case is the following. The preconditioner will scale the Laplacian equation per layer in such a way that the rows with small elements at the diagonal will get elements of order 1. However, in a shale layer we have a Neumann boundary condition at the "side" walls. But at the top and bottom we have a sandstone layer. Since the permeability in sandstone is much larger than that in shale, the pressure in the sandstone may be considered more or less constant. So from the view of a shale layer we have a kind of Dirichlet boundary condition for the top and bottom. On the other hand, for the sandstone layers, the shale layers may be regarded as more or less impermeable. The interface condition is approximately a Neumann boundary condition. So for each sandstone layer between two shale layers we have to solve a Laplacian equation with approximately Neumann boundary conditions. Only at the top layer do we have a given Dirichlet boundary condition. Since the solution of the Neumann problem is fixed up to an arbitrary additive constant we way expect a small eigenvalue for each sandstone layer that has no explicit Dirichlet boundary conditions. So it is reasonable to expect three small eigenvalues in this particular example. A mathematical proof of this observation for a diagonal preconditioner is given below.

Let our rectangular region consist of a sequence of $2n+1$ plain layers of equal thickness with a sand layer at the top and alternating shale and sand layers further down. The permeability of the sand and shale layers is 1 respectively $\epsilon > 0$. We choose a rectangular mesh with a uniform mesh size h in both x and y direction such that the sand/shale interfaces coincide with element boundaries. We discretize (1) by applying the standard first order bilinear FE-method and numerical integration with the element corner-points as the integration points. Numbering the unknowns locally from left to right and top to bottom, the element matrix S for a single element in the sand is

$$S = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}.$$

The element matrix in shale is ϵS.

Assembling all the element matrices leads to a system of equations $Ax = b$, where A is a symmetric $m \times m$ M-matrix. If we group the equations and unknowns belonging to a single layer together, add the unknowns and equations of each sand/shale interface to the corresponding sand group and order the groups from top to bottom, A becomes a block tridiagonal matrix, for which we introduce the following notation:

$$
\begin{pmatrix}
H_0 & I_1^{\mathrm{T}} & & & & & & \\
I_1 & L_1 & J_1^{\mathrm{T}} & & & & & \\
 & J_1 & H_1 & I_2^{\mathrm{T}} & & & & \\
 & & I_2 & L_2 & J_2^{\mathrm{T}} & & & \\
 & & & J_2 & \ddots & & & \\
 & & & & \ddots & & & \\
 & & & & & \ddots & & \\
 & & & & & & L_n & J_n^{\mathrm{T}} \\
 & & & & & & J_n & H_n
\end{pmatrix}
\begin{matrix}
\text{Sand + interface} \\
\text{Shale} \\
\text{Sand + interface} \\
\text{Shale} \\
\text{Sand + interface} \\
\\
\\
\\
\text{Shale} \\
\text{Sand + interface}
\end{matrix}
$$

S	S	S	S	S		S	S
a	h	a	h	a		h	a
n	a	n	a	n		a	n
d	l	d	l	d		l	d
	e		e			e	

Let Δ_h^{N}, Δ_h^{DT}, and Δ_h^{D} be the FE-matrices of the Laplacian on a single layer with respectively Neumann boundary conditions on all boundaries, Dirichlet boundary conditions on the top boundary, and Dirichlet boundary conditions on top and bottom boundaries. The matrices H_i for $i \geq 1$ consist of two parts: the contribution of the elements in the sand and the contribution of the neighboring elements in the shale. The sand part is equal to Δ_h^{N}, and the only nonzero entries in the shale part are the diagonal elements for the interface unknowns and the entries which relate neighboring unknowns in the interface. Their values are respectively 2ϵ (ϵ on the boundary) and $-\epsilon/2$. H_1 is equal to H_i apart from the entries relating to the given pressures at the top boundary, which have been eliminated. The matrices L_i are equal to $\epsilon\Delta_h^{\mathrm{D}}$. I_i and J_i have only nonzero entries on the diagonal which relates interface unknowns to their neighboring shale unknown. Their values are $-\epsilon$ ($-\epsilon/2$ on the boundary).

Let D be the diagonal of A and let $\hat{A} = D^{-1/2}AD^{-1/2}$ be the diagonally scaled matrix. Similarly to A, D can be partitioned into submatrices D_i^H, the diagonal of H_i, and D_i^L, the diagonal of L_i. The diagonal elements of D_i^H are 4 (2 on the boundary) except for the interface nodes where the values are $(2 + 2\epsilon)$ ($(1 + \epsilon)$ on the boundary). The diagonal elements of D_i^L are 4ϵ (2ϵ on the boundary). $\hat{A}$ can be partitioned into submatrices $\hat{H}_i$, $\hat{L}_i$, $\hat{I}_i$, and $\hat{J}_i$ with

$$
\begin{aligned}
\hat{H}_i &= \left(D_i^H\right)^{-1/2} H_i \left(D_i^H\right)^{-1/2}, \\
\hat{L}_i &= \left(D_i^L\right)^{-1/2} L_i \left(D_i^L\right)^{-1/2}, \\
\hat{I}_i &= \left(D_i^L\right)^{-1/2} I_i \left(D_{i-1}^H\right)^{-1/2}, \\
\hat{J}_i &= \left(D_i^H\right)^{-1/2} J_i \left(D_i^L\right)^{-1/2}.
\end{aligned}
$$

To study the influence of the parameter ϵ on the eigenvalues $\{\lambda_j^{\hat{A}}\}_{1\le j\le m}$ of the scaled matrix $\hat{A}$, we split $\hat{A}$ into an ϵ-dependent and an ϵ-independent part,

$$\hat{A} = \hat{\Delta} + \mathcal{E}, \tag{2}$$

where $\hat{\Delta}$ is the block-diagonal matrix with as first block $\hat{\Delta}_h^{\mathrm{DT}}$, the diagonally scaled Δ_h^{DT}, and then further down alternatingly $\hat{\Delta}_h^{\mathrm{D}}$, the scaled Δ_h^{D}, and $\hat{\Delta}_h^{\mathrm{N}}$, the scaled Δ_h^{N}. The eigenvalues $\lambda_j^{\hat{\Delta}}$ of $\hat{\Delta}$ are equal to the eigenvalues of all its diagonal blocks. Let $\{\lambda_j^{\mathrm{DT}}\}_{1\le j\le m_1}$, $\{\lambda_j^{\mathrm{D}}\}_{1\le j\le m_2}$, and $\{\lambda_j^{\mathrm{N}}\}_{1\le j\le m_3}$ be the ordered eigenvalues of respectively $\hat{\Delta}_h^{\mathrm{DT}}$, $\hat{\Delta}_h^{\mathrm{D}}$, and $\hat{\Delta}_h^{\mathrm{N}}$. It is well known that

$$\lambda_1^{\mathrm{N}} = 0, \tag{3}$$

and that there exists a greatest lower bound $c2(h)$ such that

$$\begin{aligned} c2(h) \le \lambda_j^{N} &\le 2 \quad \text{for } 2 \le j \le m_3, \\ c2(h) \le \lambda_j^{DT} &\le 2 \quad \text{for } 1 \le j \le m_1, \\ c2(h) \le \lambda_j^{D} &\le 2 \quad \text{for } 1 \le j \le m_2, \end{aligned} \tag{4}$$

This bound is used to separate the very small eigenvalues from the rest of the spectrum. The blocks of the symmetric tridiagonal block matrix $\mathcal{E}$ are given by

$$\begin{aligned} \mathcal{E}_{1,1} &= \hat{H}_0 - \hat{\Delta}_h^{\mathrm{DT}} \quad \text{and for } 1 \le i \le n, \\ \mathcal{E}_{2i,2i-1} &= \hat{I}_i, \\ \mathcal{E}_{2i,2i} &= \hat{L}_i - \hat{\Delta}_h^{\mathrm{D}}, \\ \mathcal{E}_{2i+1,2i} &= \hat{J}_i, \\ \mathcal{E}_{2i+1,2i+1} &= \hat{H}_i - \hat{\Delta}_h^{\mathrm{N}}. \end{aligned} \tag{5}$$

For $0 \le i \le n$, $\mathcal{E}_{2i+1,2i+1}$ contain only nonzero matrix entries which relate nodes in the interface to their neighboring interior nodes. Their values are

$$\frac{-1+\sqrt{1+\epsilon}}{2\sqrt{2+2\epsilon}} = O(\epsilon). \tag{6}$$

For $1 \le i \le n$, $\mathcal{E}_{2i,2i} = 0$ and $\mathcal{E}_{2i-1,2i}$, $\mathcal{E}_{2i,2i-1}$, $\mathcal{E}_{2i,2i+1}$, and $\mathcal{E}_{2i+1,2i}$ have only nonzero elements on the off-diagonal, relating interface nodes to their direct neighbors in the low permeable layer. The values of these entries are

$$\frac{-\sqrt{\epsilon}}{2\sqrt{2+2\epsilon}} = O(\sqrt{\epsilon}). \tag{7}$$

Let Q be the block-diagonal orthogonal matrix such that $Q^{\mathrm{T}}\hat{\Delta}Q = \Lambda^{\hat{\Delta}}$, and let B be a block diagonal matrix the blocks of which are defined by

$$\begin{aligned} B_{2i+1,2i+1} &= (\sqrt{\epsilon}/c3)I \quad \text{for } 0 \le i \le n, \\ B_{2i,2i} &= I \qquad\qquad\quad \text{for } 1 \le i \le n, \end{aligned} \tag{8}$$

where $c3$ is an arbitrary constant. If we now define $\bar{A} = B^{-1}Q^{\mathrm{T}}\hat{A}QB$ then

$$\bar{A} = B^{-1}Q^{\mathrm{T}}\hat{\Delta}QB + B^{-1}Q^{\mathrm{T}}\mathcal{E}QB = \bar{\Delta} + \bar{\mathcal{E}}. \tag{9}$$

The blocks of $\bar{\Delta}$ just contain the eigenvalues of $\hat{\Delta}$ (which satisfy Eq. (3) and inequalities (4)) and for the blocks of $\bar{\mathcal{E}}$ we find that

$$\begin{aligned} \text{the elements of } \bar{\mathcal{E}}_{2i+1,2i+1} &= O(\epsilon), \\ \text{the elements of } \bar{\mathcal{E}}_{2i-1,2i} \text{ and } \bar{\mathcal{E}}_{2i+1,2i} &= O(c3), \\ \text{the elements of } \bar{\mathcal{E}}_{2i,2i+1} \text{ and } \bar{\mathcal{E}}_{2i+2,2i+1} &= O(\epsilon/c3), \\ \text{the elements of } \bar{\mathcal{E}}_{2i,2i} &= 0. \end{aligned} \tag{10}$$

If we now choose $c3 < c2(h)/4$ and subsequently ϵ small enough, apply Gershgorin's theorem to $\bar{A}$, and account for the fact that each eigenvalue of $\hat{A}$ is also an eigenvalue of $\bar{A}$ and in the interval (0, 2), then

$$\begin{aligned} 0 < \lambda_j^{\hat{A}} &= O(\epsilon) \qquad \text{for } 1 \le j \le n, \\ c2(h)/2 + O(\epsilon) \le \lambda_j^{\hat{A}} &< 2 \quad \text{for } n+1 \le j \le m. \end{aligned} \tag{11}$$

This proves the following theorem:

THEOREM 3.1. *For ϵ small enough the diagonally scaled matrix $D^{-1/2}AD^{-1/2}$ has only n eigenvalues of $O(\epsilon)$, where n is the number of high-permeability layers ($\sigma = 1$, e.g. sand) lying between low-permeability layers ($\sigma = \epsilon$, e.g. shale).*

As one of the reviewers has remarked, similar results have been proved but yet not published in [15, 16].

4. THE DEFLATED ICCG METHOD

In this section we derive a deflated incomplete Choleski Conjugate Gradient method. This method can be used to solve the system of linear equations for the fluid pressure. In the previous section it was shown that the diagonal scaled matrix has only a small number of very small eigenvalues. A comparable spectrum has been observed for the IC preconditioned matrix. Deflation is used to annihilate the effect of the very small eigenvalues on the convergence of the ICCG method.

Let $Ax = b$ be the system of equations to be solved, where A is a symmetric and positive definite (SPD) matrix. Let M be the incomplete Choleski decomposition of A satisfying $A \approx LL^{\mathrm{T}} = M$, where L is a sparse lower triangular matrix and M is SPD. ICCG consists of the application of CG to the preconditioned system

$$L^{-1}AL^{-\mathrm{T}}y = L^{-1}b, \quad x = L^{-\mathrm{T}}y, \text{ where } L^{-\mathrm{T}} = (L^{-1})^{\mathrm{T}}.$$

Define $\tilde{A} = L^{-1}AL^{-\mathrm{T}}$ and $\tilde{b} = L^{-1}b$. Note that $\tilde{A}$ is SPD.

To define the Deflated ICCG method we assume that the vectors $v_1, \ldots, v_n$ are given and form an independent set. These vectors define a space $\mathcal{V} = \text{span}\{v_1, \ldots, v_n\}$ and a matrix $V = [v_1 \ldots v_n]$. A special choice for v_i is the eigenvectors corresponding to the smallest eigenvalues of $\tilde{A}$, hence $\tilde{A}v_i = \lambda_i v_i$, $0 < \lambda_1 \le \lambda_n \ldots \le \lambda_m$.

The operator P defined by $P = I - VE^{-1}(\tilde{A}V)^{\mathrm{T}}$ with $E = (\tilde{A}V)^{\mathrm{T}}V$ is a projection with the following properties (the matrix $E \in \mathbb{R}^{n\times n}$ is symmetric and positive definite):

THEOREM 4.1. *The operator P has the following properties*:

(i) $PV = 0$ *and* $P^{\mathrm{T}}\tilde{A}V = 0$,

(ii) $P^2 = P$,
(iii) $\tilde{A}P = P^{\mathrm{T}}\tilde{A}$.

Proof. Properties (i) and (iii) are easily checked. The proof of (ii) runs as follows:

$$\begin{aligned} P^2 &= (I - VE^{-1}(\tilde{A}V)^{\mathrm{T}})(I - VE^{-1}(\tilde{A}V)^{\mathrm{T}}) \\ &= P - VE^{-1}(\tilde{A}V)^{\mathrm{T}} + VE^{-1}(\tilde{A}V)^{\mathrm{T}}VE^{-1}(\tilde{A}V)^{\mathrm{T}} = P. \quad \blacksquare \end{aligned}$$

COROLLARY 4.1. *The matrix $\tilde{A}P$ is symmetric and positive semi-definite.*

Remark 4.1. When v_i are eigenvectors of $\tilde{A}$ with norm equal to 1 then $P = I - VV^{\mathrm{T}}$ because $v_i^{\mathrm{T}} v_j = \delta_{ij}$.

We assume that the start vector x_0 is zero, otherwise the Deflated ICCG algorithm should be applied to $A(x - x_0) = b - Ax_0$. To speed up the convergence of ICCG we assume that the space $\mathcal{V}$ is chosen such that it contains the slow converging components and split the vector y into two parts

$$y = (I - P)y + Py. \tag{12}$$

The first part $(I - P)y$ is the component of y contained in $\mathcal{V}$, whereas the second part Py is perpendicular to $\mathcal{V}$ in the $(.,.)_{\tilde{A}}$ inner product. The first part is determined from

$$(I - P)y = VE^{-1}(\tilde{A}V)^{\mathrm{T}}y = VE^{-1}V^{\mathrm{T}}\tilde{b}. \tag{13}$$

$(I - P)y$ is cheaply computable because the dimensions of $E(n \times n)$ are much less than the dimensions of $A(m \times m)$. To compute the second part Py we use $\tilde{A}Py = P^{\mathrm{T}}\tilde{A}y = P^{\mathrm{T}}\tilde{b}$ and solve y from

$$P^{\mathrm{T}}\tilde{A}y = P^{\mathrm{T}}\tilde{b}. \tag{14}$$

The singular system (14) has a solution because $P^{\mathrm{T}}\tilde{b}$ is an element of the Range $(P^{\mathrm{T}}\tilde{A})$. A solution y of (14) may contain an arbitrary element of Null $(P^{\mathrm{T}}\tilde{A}) = \mathcal{V}$. Since $PV = 0$, Py is uniquely determined.

When we apply the CG algorithm to the symmetric positive semi-definite system (14) we get the deflated ICCG algorithm:

DICCG1.

$k = 0$, $y_0 = 0$, $\tilde{p}_1 = \tilde{r}_0 = P^{\mathrm{T}}L^{-1}b$,
while $\|\tilde{r}_k\|_2 > \varepsilon$ **do**
  $k = k + 1$;
  $\alpha_k = \frac{(\tilde{r}_{k-1}, \tilde{r}_{k-1})}{(\tilde{p}_k, P^T\tilde{A}\tilde{p}_k)}$;
  $y_k = y_{k-1} + \alpha_k \tilde{p}_k$;
  $\tilde{r}_k = \tilde{r}_{k-1} - \alpha_k P^{\mathrm{T}}\tilde{A}\tilde{p}_k$;
  $\beta_k = \frac{(\tilde{r}_k, \tilde{r}_k)}{(\tilde{r}_{k-1}, \tilde{r}_{k-1})}$;
  $\tilde{p}_{k+1} = \tilde{r}_k + \beta_k \tilde{p}_k$;
end while

In order to get an approximation of y $(=L^{\mathrm{T}}x)$ the vector y_k is multiplied by P and substituted into (12).

In order to determine the matrix V we have to compute (or approximate) the eigenvectors of the matrix $\tilde{A}$. Unfortunately these eigenvectors contain many nonzero elements in our

application. Furthermore they are hard to predict on physical grounds. The eigenspace of the matrix $L^{-T}L^{-1}A$ corresponding to the n smallest eigenvalues can be approximated by the span of n vectors, which are obtained on physical grounds. In Section 5 it appears that these vectors have only nonzero elements in one high-permeability layer and its neighboring low-permeability layers. For that reason we rewrite the deflated ICCG algorithm as follows:

Define $\bar{P} = L^{-T}PL^{T}$, $\tilde{r}_k = P^{T}L^{-1}r_k = L^{-1}\bar{P}^{T}r_k = L^{-1}\hat{r}_k$, with $\hat{r}_k = \bar{P}^{T}r_k$, and $z_k = L^{-T}L^{-1}\hat{r}_k$. Since $y_k = L^{T}x_k$ and $L^{T}x_k = L^{T}x_{k-1} + \alpha_p\tilde{p}_k$, we choose $\tilde{p}_k = L^{T}p_k$. Substitution in DICCG1 leads to

DICCG2.
$k = 0, \hat{r}_0 = \bar{P}^{T}r_0,\ p_1 = z_1 = L^{-T}L^{-1}\hat{r}_0;$
while $\|\hat{r}_k\|_2 > \varepsilon$ **do**
$k = k + 1;$
$\alpha_k = \frac{(\hat{r}_{k-1}, z_{k-1})}{(p_k, P^{T}Ap_k)};$
$x_k = x_{k-1} + \alpha_k p_k;$
$\hat{r}_k = \hat{r}_{k-1} - \alpha_k \bar{P}^{T}Ap_k;$
$z_k = L^{-T}L^{-1}\hat{r}_k;$
$\beta_k = \frac{(\hat{r}_k, z_k)}{(\hat{r}_{k-1}, z_{k-1})};$
$p_{k+1} = z_k + \beta_k p_k;$
end while

It is easy to verify that the projection $\bar{P} = L^{-T}PL^{T}$ has the following properties:

Properties of $\bar{P}$.

1. $\bar{P} = I - \bar{V}E^{-1}(A\bar{V})^{T}$ where $\bar{V} = L^{-T}V$ and $E = (\tilde{A}V)^{T}V = (A\bar{V})^{T}\bar{V}$,
2. $\bar{P}\bar{V} = 0$, and $\bar{P}^{T}A\bar{V} = 0$,
3. $\bar{P}^{T}A = A\bar{P}$.

The vector x can be split into two parts (compare Eq. (12)):

$$x = (I - \bar{P})x + \bar{P}x. \tag{15}$$

The first part can be calculated as

$$(I - \bar{P})x = \bar{V}E^{-1}\bar{V}Ax = \bar{V}E^{-1}\bar{V}^{T}b.$$

For the second part we project the solution x_k obtained from DICCG2 to $\bar{P}x_k$.

For the special choice that v_i are eigenvectors of $\tilde{A}$, $\bar{v}_i$ are eigenvectors of $L^{-T}L^{-1}A$. In that case the projection can be written as $\bar{P} = I - \bar{V}(LL^{T}\bar{V})^{T}$.

A well known convergence result for CG applied to $\tilde{A}y = \tilde{b}$ is [25, p. 187]

$$\|y - y_k\|_{\tilde{A}} \le 2\|y - y_0\|_{\tilde{A}}\left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1}\right)^{k}, \tag{16}$$

where $K = K_2(\tilde{A}) = \lambda_m/\lambda_1$. Since the results obtained from DICCG1 and DICCG2 are equal we restrict our convergence research to DICCG1. When we choose $V = [v_1 \dots v_n]$ where v_i are the normalized eigenvectors of $\tilde{A}$, it is easy to verify that

$$P^{T}\tilde{A}v_i = 0 \qquad \text{for } i = 1, \dots, n,$$
$$P^{T}\tilde{A}v_i = \lambda_i v_i \quad \text{for } i = n+1, \dots, m.$$

On the space span$\{v_{n+1}, \ldots, v_m\}$ the norm $\|\cdot\|_{P^T\tilde{A}} = \|\cdot\|_{\tilde{A}P}$ is well defined. Using this norm together with inequality (16) it can be proved that

$$\|Py - Py_k\|_2 \leq 2\sqrt{K}\|Py - Py_0\|_2 \left(\frac{\sqrt{K}-1}{\sqrt{K}+1}\right)^k,$$

where $K = \lambda_m/\lambda_{n+1}$. This inequality predicts a speed up of convergence when $\lambda_n \ll \lambda_{n+1}$.

Since Py and Py_k are elements of span$\{v_{n+1}, \ldots, v_m\}$ the following expression holds:

$$\|P^T\tilde{b} - P^T\tilde{A}y_k\|_2 = \|\tilde{A}Py - \tilde{A}Py_k\|_2 \geq \lambda_{n+1}\|Py - Py_k\|_2. \tag{17}$$

So the following termination strategy can be used: estimate λ_{n+1} via Kaasschieter's method [22] and terminate DICCG1 when

$$\|P^T\tilde{b} - P^T\tilde{A}y_k\|_2 \leq \frac{\varepsilon}{\lambda_{n+1}}.$$

This together with inequality (17) implies that $\|Py - Py_k\|_2 \leq \varepsilon$.

In Kaasschieter [22] the termination criterion is derived for the unpreconditioned system $Ax = b$. It is easy to generalize this to the preconditioned system $L^{-1}AL^{-T}y = L^{-1}b$. In DICCG2 the criterion is also used when CG is applied to the left preconditioned system $M^{-1}Ax = M^{-1}b$, where $M = L^TL$. A similar termination strategy can be applied using the (in)equalities

$$\|x - x_k\|_2 = \|A^{-1}MM^{-1}r_k\|_2 \leq \|(M^{-1}A)^{-1}\|_2\|M^{-1}r_k\|_2 \leq \frac{1}{\lambda_{\min}}\|M^{-1}r_k\|_2.$$

Since $\sigma(M^{-1}A) = \sigma(L^{-1}AL^{-T})$, Kaasschieter's procedure to estimate the smallest eigenvalue can be used, which leads to the stopping criterion $\|M^{-1}r_k\|_2 < \lambda_{\min}\varepsilon$.

5. A CHOICE OF PROJECTION VECTORS

A good choice of the projection vectors is important to obtain an efficient Deflated ICCG method. In this section we restrict ourselves to the class of problems defined in Section 2. An analysis of the matrix (Section 3) shows that the spectrum of this matrix contains many small eigenvalues (of order 10^{-7}). For the preconditioned matrix, the number of small eigenvalues is drastically reduced. This number is proportional to the amount of sandstone layers. In Section 4 a Deflated ICCG method is given, which is very suitable to problems where the matrix has a small number of extreme eigenvalues.

We consider the problem as shown in Fig. 1. As a first choice we take v_1, v_2, v_3 equal to the three eigenvectors of $\tilde{A}$ corresponding to the small eigenvalues. We use DICCG1 with $P = I - VV^T$. The vectors v_i should be stored, so $3m$ extra memory positions are needed. Furthermore in every iteration of DICCG1, the projection P should be applied to a vector, which costs three inner products and three vector updates extra per iteration.

Drawbacks of this choice are:

1. the determination of the eigenvectors can be expensive,
2. the amount of extra memory and work per iteration grows, when the number of small eigenvalues increases.

For the determination of the eigenvectors an inverse (Krylov) iteration can be used, however this costs more work than the solution of the original system. In our application the fluid

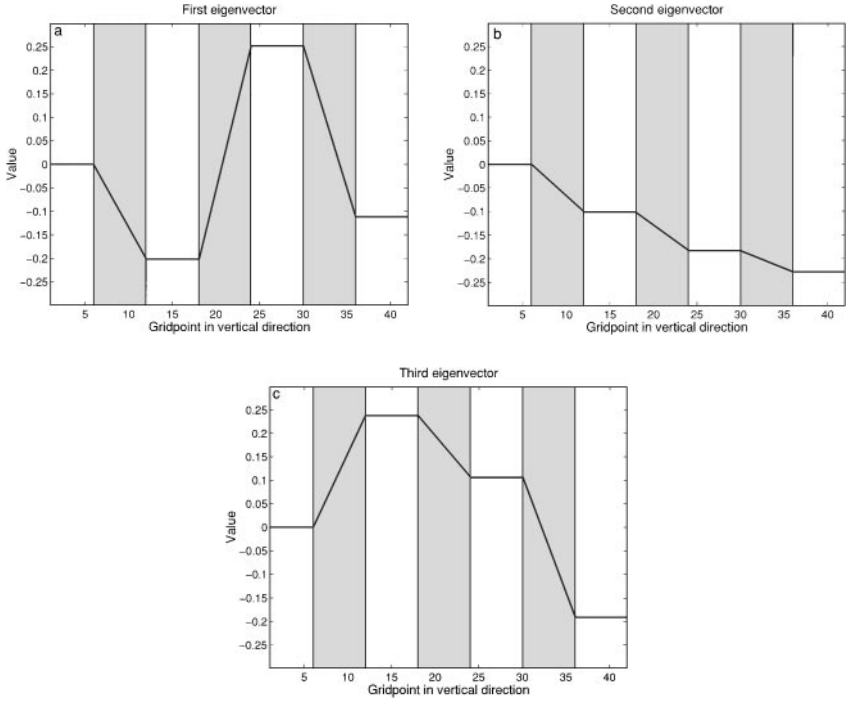

FIG. 4. The vertical cross section of the eigenvectors corresponding to the small eigenvalues.

pressure is needed in every time iteration. The differences in the matrices in consecutive time-steps are relatively small. In such a problem DICCG1, with eigenvectors as projection vectors, can be feasible when the eigenvectors are only computed at a small number of time steps.

Because of these drawbacks we use another approach, motivated by the properties of the eigenvectors $\overline{v}_i = L^{-1}v_i$ of $L^{-T}L^{-1}A$, corresponding to the small eigenvalues. For the problem considered a vertical cross section of the eigenvectors is plotted in Fig. 4. The cross-sections have the following properties:

—their value is constant in sandstone layers,
—their value is zero in the first sandstone layer,
—in the shale layers their graph is linear.

So the space span$\{\overline{v}_1, \overline{v}_2, \overline{v}_3\}$ is identical to the space span$\{w_1, w_2, w_3\}$, where the vertical cross sections of w_i are defined by

—the value of w_i is one in the $i+1$th sandstone layer and zero in the other sandstone layers,
—their graph is continuous in the whole domain and linear in the shale layers.

So instead of DICCG1 with the eigenvectors, DICCG2 is applied with $\mathcal{V} = \text{span}\{w_1, w_2, w_3\}$. Since the vectors w_i are no eigenvectors it is necessary to store w_i and Aw_i. Due to the sparseness two memory vectors are sufficient to store all w_i. Furthermore, the elements of Aw_i are only nonzero at the grid points connected to the interfaces of the ith shale layer. Thus two memory vectors are also sufficient to store all vectors Aw_i. In the same way the sparseness can be used to save CPU time. It is possible to implement the projection so that the extra amount of work per iteration is less than two inner products and two vector updates

independent of the number of small eigenvalues. This makes the DICCG2 algorithm very attractive for this kind of problems.

We have also solved problems where shale and sandstone layers are slightly curved. Again DICCG2, with projection vectors defined in the same way as above, proved to be an efficient solution algorithm. If we assume that the sandstone layers without a Dirichlet condition are numbered from 1 to n, then we propose to use DICCG2 with the projection vectors w_i chosen as:

—the value of w_i is one in the ith sandstone layer and zero in the other sandstone layers,
—in the shale layers, w_i satisfies

$$-\text{div}(\sigma \nabla w_i) = 0, \tag{18}$$

and on the interfaces it satisfies a Dirichlet boundary condition equal to the constant value 0 or 1 of the neighboring sandstone layer.

For our original problem, this choice leads to the same projection vectors as before. The solution of (18) amounts to solving the same system of equations at a much smaller domain without the extreme contrasts in the coefficients. In fact this process is similar to a domain decomposition method (compare [12, 16]).

6. NUMERICAL EXPERIMENTS

In order to test the Deflated ICCG method we have applied DICCG2 to the seven straight layers problem defined in Section 2. The three projection vectors are defined as in the previous section. For this straight layers case these vectors span exactly the space of the three eigenvectors corresponding to the small eigenvalues. Figure 5 shows the convergence behavior of the DICCG2 method, the estimate of the smallest eigenvalue as well as the true error. To facilitate comparison the norm of the error using ICCG and DICCG2 is given in Fig. 6. Since one iteration of DICCG2 costs approximately 30% more CPU time than one

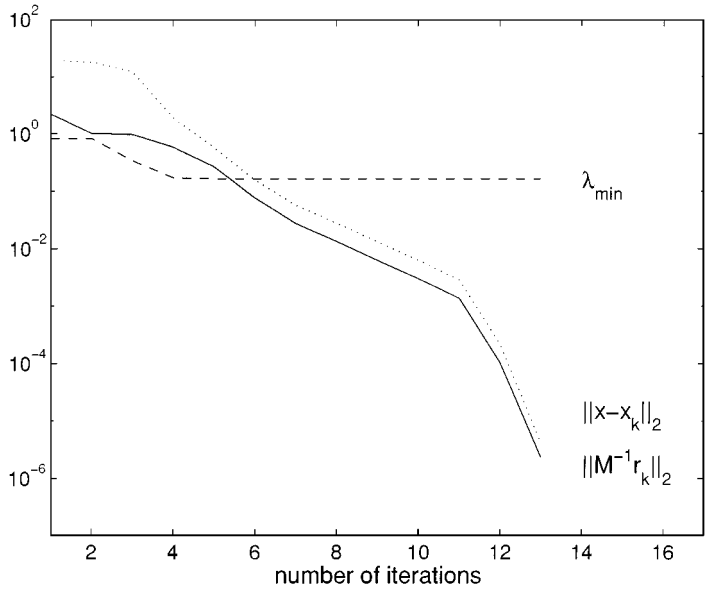

FIG. 5. Convergence behavior of DICCG2 for the straight layers problem.

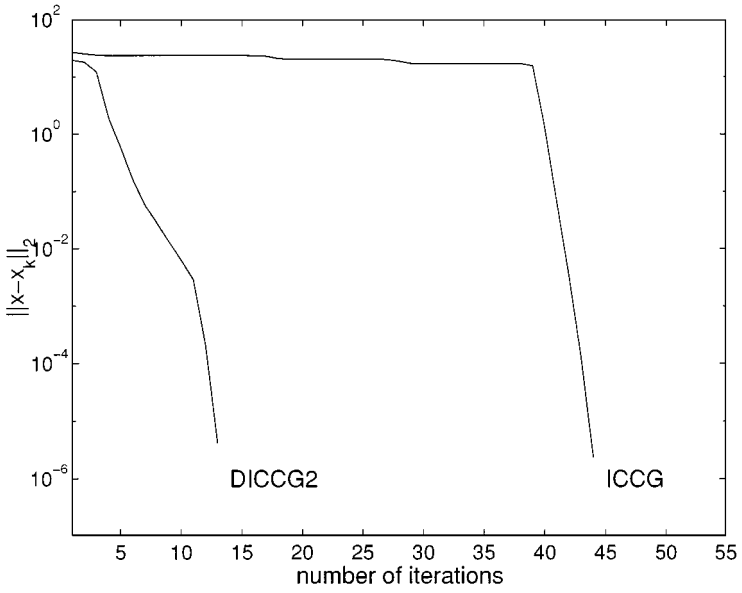

FIG. 6. Norm of the error for the straight layers problem.

iteration of ICCG, we have a large improvement when the deflated method is used. Besides that, the decrease of the residual is now a measure for the error, so that we have a reliable termination criterion.

Our intention is to use the DICCG2 method also for more complicated regions, where we only have "approximate" eigenvectors. Therefore we have replaced the straight layers in our example by curved layers as shown in Figs. 7 and 8. Both domains are a subset

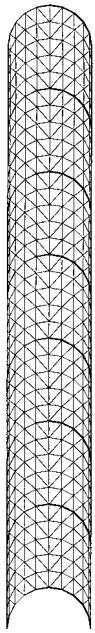

FIG. 7. Mesh used in the parallel arcs layered problem.

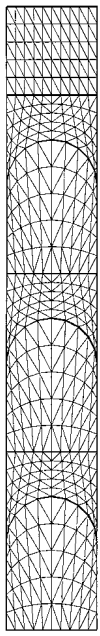

FIG. 8. Mesh used for straight and curved layers.

$\mathbb{R}^2$. The number of elements is exactly the same as for the straight layers region. For these examples the graphs of the vertical cross sections of the eigenvectors are no longer linear in the shale layers. Nevertheless we use exactly the same projection vectors in DICCG2 as for the straight layers problem. The convergence behavior of the DICCG2 method applied to the mesh of Fig. 7 is shown in Fig. 9. The number of iterations has been increased compared to the straight layers case, but the overall behavior is the same. Application of the

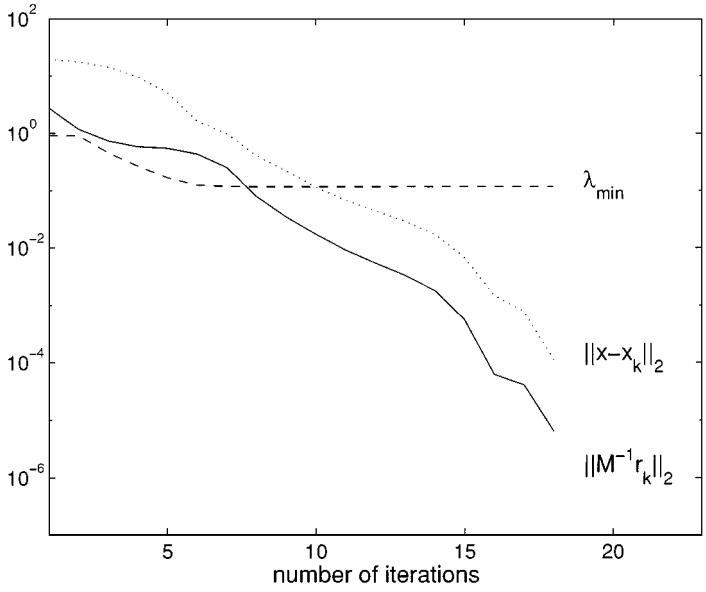

FIG. 9. Convergence behavior of DICCG2 for the parallel arcs problem.

DICCG2 method to the mesh of Fig. 8 leads to the same convergence behavior as the one for the straight layers (Fig. 5). Presumably, the projection vectors are good approximations of the eigenvectors in these cases, although the relative difference between them is of order 10^{-1}. From our limited number of experiments it is clear that the DICCG2 method is an enormous improvement compared to the classical ICCG method, provided the approximate eigenvectors are a reasonable estimate of the true eigenvectors.

7. CONCLUSIONS

It has been shown that the preconditioned Conjugate Gradient method for layered problems with extreme contrasts in the coefficients has a very erratic convergence behavior. The residual shows large bumps and moreover the decrease of the residual cannot be used as reliable termination criterion. Only when all eigenvectors corresponding to small eigenvalues are detected, which means that the smallest Ritz values are converged to the smallest eigenvalues, is the convergence behavior more or less as usual. In order to solve this problem a new method called DICCG has been developed that projects the contribution of the eigenvectors corresponding to the small eigenvalues onto the null space. This new method has excellent convergence properties and, more important, a reliable termination criterion. Even if we use approximations of these eigenvectors based on physical arguments still the deflated ICCG method performs very well.

It is our aim to apply the DICCG method to large time-dependent 3D problems with a realistic number and shape of layers. A point to be solved, however, is how to create the approximate eigenvectors in more general configurations including inclusions. We think that it is sufficient to solve the original problem for each completely enclosed shale layer with appropriate boundary conditions. Since we are only dealing with approximate eigenvectors we expect that the solution of the subproblem may be done with moderate accuracy. The choice of the approximate eigenvectors, as well as the sensitivity of the method to the accuracy of these approximate eigenvectors, is the subject of our present research.

ACKNOWLEDGMENTS

The authors thank Mohammed Mokaddam for doing numerical experiments and one of the referees for pointing out Refs. [15, 16] to us.

REFERENCES

1. R. E. Alcouffe, A. Brandt, J. E. Dendy Jr., and J. W. Painter, The multigrid method for diffusion equations with strongly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.* **2**, 430 (1981).
2. M. Arioli and D. Ruiz, Block Conjugate Gradient with subspace iteration for solving linear systems, in *Iterative Methods in Linear Algebra II, Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, Blagoevgrad, Bulgaria, June 17–20, 1995*, edited by S. D. Margenov and P. S. Vassilevski, IMACS Series in Computational and Applied Mathematics (IMACS, Piscataway, NJ, 1996), Vol. 3, p. 64.
3. O. Axelsson, *Iterative Solution Methods* (Cambridge Univ. Press, Cambridge, UK, 1994).
4. O. Axelsson and G. Lindskog, On the eigenvalue distribution of a class of preconditioning methods, *Numer. Math.* **48**, 479 (1986).
5. J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, Adaptively preconditioned GMRES algorithms, SCCM-96-15 (Stanford University, Stanford, 1996).

6. J. Bear, *Dynamics of Fluids in Porous Media* (American Elsevier, New York, 1972).
7. E. F. F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F. W. Wubs, and P. M. de Zeeuw, How fast the Laplace equation was solved in 1995, *Appl. Numer. Meth.* **24**, 439 (1997).
8. P. N. Brown and H. F. Walker, GMRES on (nearly) singular systems, *SIAM J. Matrix Anal. Appl*. **18**, 37 (1997).
9. K. Burrage, J. Erhel, and B. Pohl., A deflation technique for linear systems of equations, *SIAM J. Sci. Comp.*, 1997 (to appear).
10. A. Chapman and Y. Saad, Deflated and augmented Krylov subspace techniques, *Numer. Linear Algebra Appl.* **4**, 43 (1997).
11. R. K. Coomer and I. G. Graham, Massively parallel methods for semiconductor device modelling, *Computing* **56**, 1 (1996).
12. P. Deuflhard and K. Lipnikov, Domain decomposition with subdomain CCG for material jump elliptic problems, *East–West J. Numer. Math.* **6**, 81 (1998).
13. J. Erhel, K. Burrage, and B. Pohl, Restarted GMRES preconditioned by deflation, *J. Comp. Appl. Math.* **69**, 303 (1996).
14. Melvyn R. Giles, *Diagenesis: A Quantitative Perspective; Implications for Basin Modelling and Rock Property Prediction* (Kluwer, Dordrecht, 1997).
15. I. G. Graham and M. J. Hagger, Additive Schwarz, CG and discontinuous coefficients, in *Proceedings of 9th International Conference on Domain Decomposition*, edited by P. Bjorstad, M. Espedal, and D. Keyes (to appear).
16. I. G. Graham and M. J. Hagger, Unstructured additive Schwarz–CG method for elliptic problems with highly discontinuous coefficients (Preprint 96/08, University of Bath, Bath, 1996). [*SIAM J. Sci. Comput.* (to appear)]
17. I. Gustafsson, A class of first order factorization methods, *BIT* **18**, 142 (1978).
18. M. J. Hagger, *Iterative Solution of Large, Sparse Systems of Equations, Arising in Groundwater Flow Models*, Ph.D. thesis (University of Bath, Bath, 1995).
19. B. Heise, Parallel solvers for the FEM–BEM equations with applications to non-linear magnetic field problems, in *Numerical treatment of coupled systems, Proceedings of the Eleventh GAMM-Seminar, Kiel, January* 20–22, 1995, edited by W. Hackbusch and G. Wittum, Notes on Numerical Fluid Mechanics (Vieweg, Brannschweig, 1995), Vol. 51, p. 73.
20. M. R. Hestenes and E. Stiefel, Methods of Conjugate Gradients for solving linear systems, *J. Res. Nat. Bur. Stand.* **49**, 409 (1952).
21. Victoria E. Howle and Stephen A. Vavasis, Preconditioning complex-symmetric layered systems arising in electrical power modeling, in *Fifth Copper Mountain Conference on Iterative Methods, Copper Mountain, Colorado, March 30–April 3, 1998*, edited by T. A. Manteuffel and S. F. McCormick (1998).
22. E. F. Kaasschieter, A practical termination criterion for the Conjugate Gradient method, *BIT* **28**, 308 (1988).
23. E. F. Kaasschieter, Preconditioned Conjugate Gradients for solving singular systems, *J. Comp. Appl. Math.* **24**, 265 (1988).
24. S. A. Kharchenko and A. Yu. Yeremin, Eigenvalue translation based preconditioners for the GMRES(k) method, *Numer. Lin. Alg. Appl.* **2**, 51 (1995).
25. D. G. Luenberger, *Introduction to Linear and Nonlinear Programming* (Addison–Wesley, New York, 1973).
26. J. A. Meijerink and H. A. Van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comput.* **31**, 148 (1977).
27. R. B. Morgan, A restarted GMRES method augmented with eigenvectors, *SIAM J. Matrix Anal. Appl.* **16**, 1154 (1995).
28. S. G. Mulyarchik, S. S. Bielawski, and A. V. Popov, Efficient computational schemes of the Conjugate Gradient method for solving linear systems, *J. Comput. Phys.* **110**, 201 (1994).
29. R. A. Nicolaides, Deflation of Conjugate Gradients with applications to boundary value problems, *SIAM J. Numer. Anal.* **24**, 355 (1987).
30. Y. Notay, Incomplete factorizations of singular linear systems, *BIT* **29**, 682 (1989).
31. Y. Notay, DRIC: A dynamic version of the RIC method, *J. Numer. Linear Algebra* **1**, 511 (1994).

32. Y. Saad, ILUM: A multi-elimination ILU preconditioner for general sparse matrices, *SIAM J. Sci. Comput.* **17**, 830 (1996).
33. Y. Saad, Analysis of augmented Krylov subspace methods, *SIAM J. Matrix Anal. Appl.* **18**, 435 (1997).
34. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
35. Y. Saad, M. Yeung, J. Ehrel, and F. Guyomarc'h, *A Deflated Version of the Conjugate Gradient Algorithm*, Technical Report UMSI 98/97 (Department of Computer Science and Engineering, University of Minnesota, Minneapolis, 1998).
36. G. Shroff and H. B. Keller, Stabilization of unstable procedures: The recursive projection method, *SIAM J. Numer. Anal.* **30**, 1099 (1993).
37. A. van der Ploeg, E. F. F. Botta, and F. W. Wubs, Nested grids ILU-decomposition (NGILU), *J. Comput. Appl. Math.* **66**, 515 (1996).
38. A van der Sluis and H. A. van der Vorst, The rate of convergence of Conjugate Gradients, *Numer. Math.* **48**, 543 (1986).
39. H. van der Veen, K. Vuik, and R. de Borst, Post-bifurcation behavior in soil plasticity: Eigenvector perturbation compared to deflation, in *Computational Plasticity; Fundamentals and Applications, Part 2, Proceedings of the Fifth International Conference on Computational Plasticity, Barcelona, Spain, 17–20 March, 1997*, edited by D. R. J. Owen, E. Oñate, and E. Hinton, p. 1745 (CIMNE, Barcelona, 1997).
40. C. Vuik, A. Segal, and J. A. Meijerink, *An Efficient Preconditioned CG Method for the Solution of Layered Problems with Extreme Contrasts in the Coefficients*, Report 98-20, (Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, 1998).